

Online Appendix:

Behavioral Communities and the Atomic Structure of Networks

by Matthew O. Jackson and Evan C. Storms

6 A Comparison of Behavioral Communities and Modularity-Based Community Detection Algorithms

Theorem 1 provides a distinction between our approach and other community detection algorithms. Whether a particular block is identified as an atom depends on the threshold(s) Q in question. If there is heterogeneity in the relative cohesiveness of different blocks, then different blocks can be identified as atoms for different thresholds, with some blending together for some and not other thresholds. It can be that there is no threshold that identifies all the blocks simultaneously. This is particularly important, since it means that the blocks themselves might not be the right atomic partition for any given Q .

Many community detection algorithms simply return one partition, and have no analog to a behavioral threshold. In particular, it can be that the partition generated by, for instance the Louvain or other modularity based methods, never correspond to the partition corresponding to any Q . So it is not that they are identifying some robust partition, but instead that they are identifying something that never corresponds to something behaviorally generated, robust or not. For instance, if different blocks have different relative densities of internal vs external linking probabilities, it will never be the case that the partition of all blocks corresponds to a robust partition. If one includes all the q 's that isolate different blocks in one Q then the robustness ends up with a degenerate partition being the unique robust equilibrium. Other detection algorithms deliver a nested hierarchy of partitions, which the researcher can then choose between by selecting some way to value the fineness of the partition. However, those hierarchies can subdivide the blocks as they become fine, and yet not identify the blocks or actual atoms if they are coarse. Again, they can never correspond to the atoms, except at the extremes where the partition is the degenerate set of all nodes. In our approach, the partition varies in an

interpretable way with the behavioral threshold.

To illustrate this contrast, we compare the communities that are found via a most popular modular method—the Louvain method—to our atoms. That method is based on a modularity measure, and looks to identify sets of nodes that have a higher likelihood of links between them than to outside nodes (Blondel et al. (2008)). The issue is that different qs can identify different sets of nodes, and if one then tries to include the full range of all those qs , the robust atoms become degenerate. In particular, we show how for some q 's our atoms are strictly finer than those modules, while for others ours are a coarsening, and for other qs they are non-nested. This is pictured in Figure 8, which shows that behavioral atoms differ from the communities defined by standard community detection algorithms on a prominent data set (Add Health). We use the high school social network from Figure 4.

Panel (a) of Figure 8 depicts the communities identified by a modularity-minimizing algorithm (specifically, the Louvain method for community detection, Blondel et al. (2008)) via the dotted outlines and color in subfigure (a), while the colors in the other panels represent the behavioral atoms.

First, note that in comparing panels (a) and (b), we see that for the $Q = 1/2 \pm \varepsilon$ robust neighborhood of $q = 1/2$, the atoms are finer than the communities obtained from the Louvain method. The additional fineness is capturing real behavioral divides and thus are not defects in terms of defining atoms but real information that behavioral atoms convey that other methods do not. In particular, it is possible to have conventions that would cut across the Louvain communities.

Next, note that in comparing panels (a) and (c), we see that for the $Q = 1/3 \pm \varepsilon$ robust neighborhood of $q = 1/3$, the atoms sometimes subdivide the communities from the Louvain method, and sometimes cut across them, and sometimes doing both at the same time; e.g., the pink and white nodes in the two communities in the upper left hand corner.

For behaviors with low thresholds, contagion occurs across blocks. For this network, a behavior that has a $q = 1/4$ threshold cannot be contained by any blocks, and the atomic structure reflects that as we see in panel (d).

An interesting feature of Figure 8 is the orange atom in panel (b) involves nodes that are not directly connected to each other. Here, the orange nodes nonetheless form an atom because they have similar shares of their neighbors in the white, pink, and green atoms, so that any conventions involving these larger atoms includes either all or none of the orange nodes. The Louvain method ends up splitting them because it is based only on modularity and not how nodes would behave, and by contrast our method identifies them as an atom that will always act the same in any convention. Again, we emphasize an important feature of our approach: the community structure varies with the behavior, and we can see different atoms and blocks emerge at different thresholds, some of which are not discovered at all by standard methods. Modular methods can thus be poor predictors of the possible norms of behavior.

7 Atom-Finding Algorithm: Details

Here we provide a more detailed description of our approximation algorithm for finding the $Q = [\underline{q}, \bar{q}]$ atomic structure of a network G . As discussed in section 8, the high-level approach is to first generate conventions by finding the Q -conventions generated by small (up to size k) subsets of nodes, and then using an ILP solver to test whether as of yet unseparated nodes have separating conventions not found in the first step.

If we choose k such that the Q -atomic partition is k -generatable, the second step is unnecessary: the first step will always find the correct partition. However, since in practice we do not know for what k the partition is k -generatable, we can modify the first step so that we find conventions even when they are not necessarily k -seedable, by adding additional nodes to a subset when the convention it generates is empty. We select which nodes to add by whichever most increase the cohesion of the subsets $\underline{q}$ -closure. Note that if we can generate a non-trivial convention in this way, we can generate that convention starting from a connected subsets of nodes; hence with this modification we can restrict the subsets we attempt to grow into conventions to connected size- k subsets.

Algorithm 1 (Generating the Q -Atomic Partition).

1. Generate all connected subsets of G up to size k .
2. For each subset S generated above, compute the Q -convention generated by S . If this is the empty convention, add to S the node which most increases the cohesion of the $\underline{q}$ -closure of S , and repeat with the augmented subset in place of S . Let $\mathcal{C}$ be the collection of Q -conventions generated this way, and let P be the set of pairs of nodes not separated by some convention in $\mathcal{C}$.
3. While P is nonempty, select a pair of nodes $(i, j) \in P$, and solve the ILP i, j $Q - SPLIT$. If a separating convention is found, add this convention to $\mathcal{C}$ and remove from P all pairs which the convention separates. If no separating convention is found, remove (i, j) from P .
4. Return the partition generated by the conventions in $\mathcal{C}$.

Because step 3 involves solving an ILP, there is no polynomial-time method for ensuring a correct solution (i.e. identifying a Q -convention if it exists). We rely on ILP solvers³ to find a correct solutions with high probability; the accuracy results reported in Section 8 suggest that the solver is in fact finding the correct solution with high enough probability to generate the correct atomic structure in the vast majority of real-world networks in our test set.

Code implementing the above algorithm can be found in the supplementary material.

³Reported results use Gurobi, one of the most widely-used combinatorial solvers. Accuracy is similar with the open source solver Cbc, though the time taken to compute a solution is on average thirty percent longer.

8 Computability of the Atomic Partition

Finding a Q -convention that separates two nodes in an arbitrary network is a combinatorial set problem that, because of the complementarity of nodes' influence in spreading behavior, cannot be reduced to a submodular maximization problem. It is unsurprising, then, that the problem is NP hard—indeed, even just determining whether a network has a proper Q convention (one other than the entire network) is NP hard. We therefore cannot hope for a method that identifies all of the Q atoms for an arbitrary network in polynomial time unless $P = NP$.⁴

In this section, we present two approaches to this challenge. First, we describe a method for reducing the atom-finding problem to an integer linear program (ILP), which is known to perform well in many NP-hard problems on real-world data. Second, we discuss a local-richness condition on a network that arises naturally in many settings of interest and show that under this condition the Q atoms can be determined efficiently.

In addition, we use a variety of real-world networks to demonstrate that standard ILP solvers are able to solve the associated ILP and that the real-world networks exhibit our local-richness condition.

8.1 The i, j $Q - SPLIT$ Problem

Given a network (N, g) , two nodes $i \neq j \in N$, and a compact interval of thresholds $Q = [\underline{q}, \bar{q}]$, we term the problem of determining whether there exists a Q -convention in g which contains exactly one of i and j the i, j $Q - SPLIT$ problem, and refer to such a convention as a **separating** Q -convention for i and j . Note that finding the Q -atomic structure of g is equivalent to solving i, j $Q - SPLIT$ for each ordered pair of nodes i, j . Since there are $(n^2 - n)/2$ such pairs, the problem of finding the Q -atomic structure is efficiently reducible to i, j $Q - SPLIT$.

⁴For more background on the complexity of computing equilibria in coordination games on networks see Deligkas et al. (2023).

8.2 ILP Formulation

The i, j $Q - SPLIT$ problem admits a natural representation as an ILP. Given a network g with associated adjacency matrix G , let $\tilde{G}$ be the degree-regularized adjacency matrix obtained by dividing each row by the degree of the corresponding node. For $S \subseteq V$, let $x_S \in \{0, 1\}^n$ be a binary vector with $x_S(i) = 1$ if node i is in S . Then i, j $Q - SPLIT$ is exactly the problem of finding a feasible solution x_S to the following system of equations:

ILP Formulation $x_S(i) + x_S(j) = \mathbf{1}$ $(\tilde{G} - \bar{q}I)x_S \geq 0$ $(\tilde{G} - (1 - \mathfrak{q})I)x_{S^c} \geq 0.$

The first condition requires that the subset S contain exactly one of i and j . The second condition requires that S be $\bar{q}$ -cohesive. The third condition requires that S be $\mathfrak{q}$ -closed, expressed in the equivalent formulation that the complement of S be $1 - \mathfrak{q}$ -cohesive (where $\mathfrak{q}$ is a rational number, $\mathfrak{q}$ can be replaced with $\mathfrak{q} - \eta$ for $\eta < 1/n$ to deal with potential ties).

Although finding the solution to an ILP is NP-hard, state-of-the-art ILP engines have consistently found feasible (and near-optimal) solutions in many real-world applications in runtimes that are orders of magnitude below the theoretical worst-case. Moreover, since each solution to the i, j $Q - SPLIT$ problem separates many pairs of nodes, in practice we can determine the Q -atomic partition by solving far fewer than $n^2 - n$ instances of i, j $Q - SPLIT$. At the end of this section we demonstrate that with a standard ILP solver we can correctly identify the Q -atoms in the Indian Village, Add Health, and Caltech networks in times that grow linearly with the number of nodes.

Before showing the efficiency in the data, we describe a condition that guarantees that an ILP works efficiently.

8.3 k -seedable Conventions

Social networks exhibit clustering and sparsity, both of which help promote the existence of conventions that grow from a set of seeds much smaller than the total number of nodes.

If the collection of such conventions is rich enough to generate the atomic structure of the network, then computing the atomic structure becomes tractable.

Formally, we say that a Q -convention C is **generated** by a set of nodes $S \subset C$ if it is the one that is obtained the process we described for our seeding analysis in Section 4 when using threshold $\mathfrak{q}$. Note that since it is a Q -convention, it remains a convention for $\bar{q}$. A Q -convention C is **k -seedable** if C is generated by some $S \subset C$ with $|S| \leq k$.

Our local structural richness condition requires that k -seedable conventions generate the Q -atomic structure of g : The Q -atomic structure of g is **k -generatable** if, whenever there exists an i, j -separating Q -convention, there exists a k -seedable i, j -separating Q -convention. Note that this does not require that g have *only* k -seedable conventions, but instead that if two nodes can be separated by a Q -convention, then they can be separated by some k -seedable Q -convention.

Since there are polynomially many subsets of size less than or equal to a given k , and since the convention generated by a particular subset can be computed in $O(n^2)$ time, k -generatability of the Q -atomic structure is a sufficient condition for efficient computation. This is formally expressed as follows:

CLAIM 1. *In the class of networks with k -generatable Q -atomic structures for $k \geq 4$, i, j Q – *SPLIT* can be solved in $O(n^{k+4})$ time, and the Q atomic structure of a network can be determined in $O(n^{k+6})$ time.*

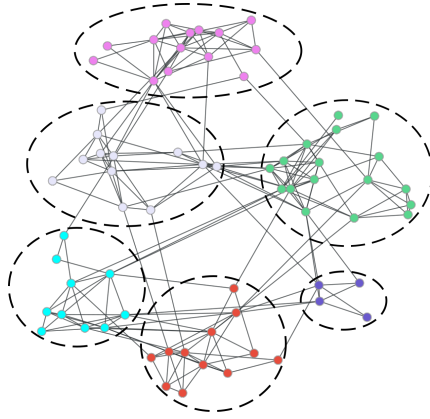
To see the proof of this claim, first observe that the $\mathfrak{q}$ -closure of a subset S can be computed by multiplying x_S by $\tilde{G}$ to determine which nodes have a share of at least $\mathfrak{q}$ of their neighbors in S , replacing x_S by $x'_{S'}$ where $S' = \{i \in N | i \in S \text{ or } \tilde{G}x_S(i) \geq \mathfrak{q}\}$, and iterating until the set of nodes does not change. This process terminates in at most n steps, and since matrix multiplication can be computed in $O(n^3)$ time, the closure can be obtained in $O(n^4)$ time. Finding the convention generated by a subset involves two closure operations, and so can also be completed in $O(n^4)$ time. Next, note that for any nodes i, j of g , if the Q atomic structure of G is k -generatable, then there exists a subset S of size at most k generating a Q -convention that separates and separable pair of nodes i and j . There are $O(n^k)$ such subsets, and so determining whether there exists an i, j -separating

Q convention can be performed in $O(n^{k+4})$ time. Finally, solving i, j $Q - SPLIT$ for each of the $(n^2 - n)/2$ pairs of nodes determines the Q -atomic partition of g , and so this partitioning can be determined in $O(n^{k+6})$ time.

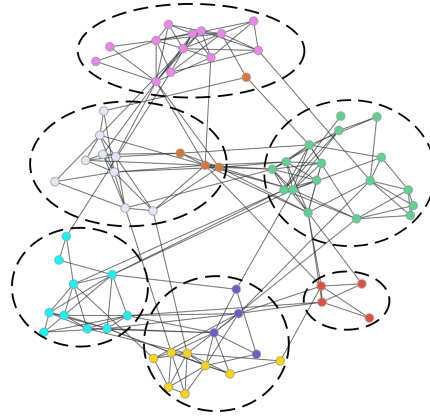
The property of a network having a k -generatable Q -atomic partition is an intuitive and natural one, but clearly determining whether a network g has a k -generatable Q -atomic partition is necessarily NP-hard. Providing sufficient conditions on a network structure for this to hold is something that we leave for future research. Still, we can verify that this property is satisfied on several data sets that have been used extensively in the literature and have nontrivial sizes to the network. In particular, our next section provides evidence that in many real-world social networks, the Q -atomic structure is k -generatable for k orders of magnitude smaller than the number of nodes, so that k -generatability for small k may be reasonable a-priori in at least applications using common data sets.

Supplementary References

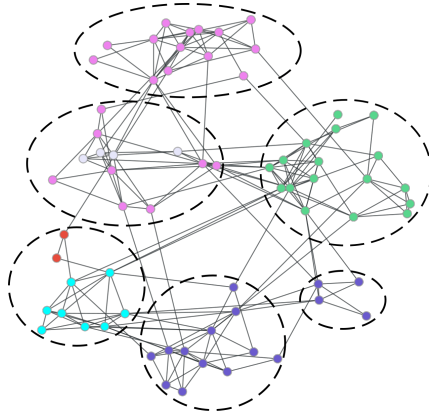
- BLONDEL, V. D., J.-L. GUILLAUME, R. LAMBIOTTE, AND E. LEFEBVRE (2008): “Fast unfolding of communities in large networks,” *Journal of statistical mechanics: theory and experiment*, 2008, P10008.
- DELIGKAS, A., E. EIBEN, G. GUTIN, P. R. NEARY, AND A. YEO (2023): “Some coordination problems are harder than others,” *arXiv preprint arXiv:2311.03195*.



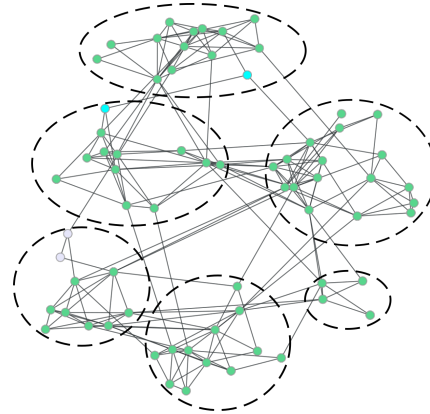
(a) The communities found via modularity minimization (colored and outlined).



(b) The $Q = 1/2 \pm \varepsilon$ robust behavioral atoms (color) compared with the modular communities (outline). 8 out of 69 = 11.6% of the nodes have to be reclassified to go from one partition to the other.



(c) The $Q = 1/3 \pm \varepsilon$ behavioral atoms (color) compared with the modular communities (outline). 14 out of 69 = 20.3% of the nodes have to be reclassified to go from one partition to the other.



(d) The $Q = 1/4 \pm \varepsilon$ robust behavioral atoms (color) compared with the modular communities (outline). 52 out of 69 = 71.1% of the nodes have to be reclassified to go from one partition to the other.

Figure 8: Comparing the behavioral atoms from various thresholds to the communities found via the Louvain method of modularity minimization (Blondel et al (2008)). Dotted outlines and color in subfigure (a) are the Louvain method communities, while the colors in the other panels represent the behavioral atoms for various choices of thresholds. Nodes with no connections are omitted from the figures.